DIAGNOSIS: REASONING FROM FIRST PRINCIPLES AND EXPERIENTIAL KNOWLEDGE

Linda J.F. Williams and Dennis G. Lawler
McDonnell Douglas Astronautics Company
Engineering Services
16055 Space Center Blvd.
Houston, TX 77062

1. INTRODUCTION

The overall goal of this paper is to show that completeness and efficiency can be obtained when automating diagnostic reasoning systems by using a combination of two approaches; diagnosis from first principles and diagnosis from experiential knowledge.

What we mean by diagnosis from first principles is reasoning about a malfunction using design knowledge; the design knowledge being the description of the system structure and behavior that can be obtained from documentation, schematics, drawings, etc. This approach to diagnosing a malfunctioning system through the use of a deep understanding of the fundamental structure and behavior of the system and its components has the target of providing an expert's troubleshooting ability without explicitly modeling the expert (6). In contrast to this is the approach of diagnosis from experiential knowledge. Experiential knowledge is of course knowledge gained from experience; knowledge of how to troubleshoot a system or reason about a malfunctioning system using knowledge gained from a practical viewpoint. Earlier diagnostic systems such as MYCIN used this approach; (3,7) MYCIN derived its capabilities by implementing a model of an expert's experienced-based reasoning. In section 2 and 3 we discuss further the ideas of diagnosis from first principles and diagnosis from experiential knowledge.

The efficacy of diagnosis from experiential knowledge has been proved through the development of useful application programs currently being integrated into various operations and organizations. Applications that display reasoning behavior based on first principles, i.e. a 'deep knowledge' of the system, are not pervasive yet but research is very active in this area (2). First principle knowledge about the structure and behavior of common system components allows the development of library packages of knowledge for general use in many applications. Additionally, it is thought that first principle diagnostic reasoning systems will provide a more complete understanding of both system function and malfunction than the experiential reasoning approach. Both approaches have distinct advantages to offer for development of a diagnostic application and both approaches are sufficient for use in some situations; (3,7)(6) however, both approaches have disadvantages when implemented as the only diagnostic reasoning process in an automated system. These disadvantages prevent either approach from being adequate or robust enough to handle diagnostic reasoning needed for the highly complex systems being studied and developed in todays space programs. We expand our discussion of advantages and disadvantages of both approaches in section 4 and give examples showing why a combination of the two approaches is necessary to obtain a complete and efficient automated diagnostic reasoning system in section 5.

Knowledge acquisition for intelligent diagnostic reasoning systems is an issue that deserves some discussion. Although we are capable of automating diagnostic information, obtaining a 'complete set' of knowledge needed for an efficient diagnostic reasoning system can be a difficult and endless process. We feel that a systematic approach to acquiring knowledge will facilitate defining the program's scope of competence and assist in guiding the knowledge acquisition process. In section 6 we discuss knowledge acquisition relative to our observation about a combined reasoning approach to diagnosis and elaborate on what knowledge is necessary and where the knowledge is likely to be found.

2. DIAGNOSIS FROM FIRST PRINCIPLES

A complete description of diagnosis from first principles would require a much deeper discussion than is appropriate for this paper. In lieu of this we present a brief discussion based on illustrating what we feel are the major concepts of the approach and their interrelation and definitions.

Tersely, diagnosis from first principles is

reasoning solely from a description of the structure and behavior of the system and its components to explain discrepancies between observed and correct system behavior. By structure we mean a complete list of a system's components and the specification of their connectivity.

The components in this description are all the elements of the system that conceivably effect the system's behavior. Each component specification should include a description of the generic characteristics and behavior of the component as well as any additional configuration-specific characteristics and behavior peculiar to the system in question (e.g. resistance, EPROM programming, etc.). In operation the diagnostic program would use such information to determine the qualitative state of both the internal component behavior as well as the component's external environment (e.g. is it receiving appropriate input, is the component overheated, etc.).

Connectivity here means the specification of all possible connections that provide a path for a relevant effect or influence on a component. The term effect here must be taken in the broadest sense possible meaning not only physical but abstract effects such as instruction or information passing between components. Specific types of effects that are used depend on the type and resolution of the specific model being used to represent the behavior of the component in question. Where appropriate, distinction might be made between intended and unintended effects on a component and the attendant paths. Effects, of course, must be observed in some sense so that the appropriate component behavior can be computed and compared with the actual behavior exhibited by the system. Hence a set of observables must be specified which are needed for this computation. These would include all component inputs and outputs as well as the value of any internal state variables (5). When determining the health of particular component the value of every pertinent observable may not be available. In such cases a measurement may be needed. Commonly this involves the use of a specific piece of instrumentation, either in place (e.g. BIT/BITE) or provided by the expert (e.g. the use of a multimeter), or by reconfiguring the system in some sense and then remeasuring specific observables. If making a measurement is not possible, the value may have to be derived indirectly via a computation involving other observables, or by a heuristic relationship.

In addition to the above descriptive knowledge about a system, a general computational approach to diagnosis that utilizes this knowledge is necessary for a complete diagnostic program. Several of these computational approaches do exist (6) however this is still an active research area and is beyond the scope of this paper.

3. DIAGNOSIS FROM EXPERIENCE

When diagnosing a system the expert(s) have several conceptual tasks confronting them: monitoring the system observables, detecting malfunctioning behavior from this monitoring task, isolating the component(s) responsible for the behavior, and recovering from the malfunction. These tasks are usually approached with a specific strategy to optimize the task performance. The monitoring strategy (e.g. frequency of observation) is developed a priori.

The detection and isolation tasks are performed with the monitored information and with strategies whose specifics commonly change depending on the type of behavior being observed. For any reasonable size of system the strategy can become so complex as to be arbitrary to an uninformed observer of the expert's performance. As mentioned in the discussion above various means may be needed to gather information about a particular state variable and so the development of a behavior specific measurement strategy is commonly required. Additionally, a strategy for recovering the system to a safe and/or effective behavior is also required.

The above discussion of diagnosis from first principles includes many of the concepts needed to support these four general task areas but not all the considerations that are used by the expert in performing the diagnosis. Specifically missing is a discussion of those situation observables that are required for the development of the measurement and recovery strategies.

These strategy development tasks very frequently involve reasoning about extra-system factors that would not necessarily be explicitly modeled in the structure and behavior of the system itself or would be impossible to model adequately given the current understanding of the system in question. Also, this reasoning is quite likely to be context dependent, highly judgmental, and almost certainly experientially derived. The reasons for this vary but some common ones are: cost and/or design considerations limit the amount of instrumentation in the system, cost and/or design considerations limit the amount of measurement access (e.g. IC packaging), quality of information may vary (e.g. instrumentation failure), context may limit plan for measurement (e.g. safety limitations, politics or other indeterministic reasoning) may come into play.

4. COMPARISON OF APPROACHES

In order to understand why a combination of diagnosis from first principles and diagnosis from experiential knowledge is needed to automate complete and efficient diagnostic reasoning systems, we begin to explore the advantages and disadvantages of each approach.

There are several advantages in basing reasoning on first principle knowledge. A model developed from a description of structure and behavior provides a fairly complete and indepth description of the system that is to be automated. This description provides a deep understanding of the system that the expert troubleshooter will often refer to in order to compliment experiential knowledge. Since this model includes descriptions of all components incorporated in the system being modelled, there is knowledge about components that can be ported across applications. For example, if the system we wish to model is a device built from digital logic components, each digital logic component that is described can be reused when modeling a system that requires the same component (2). Since we can describe every component in a system, irregardless of the level of detail, a library of component descriptions can be built to facilitate construction of new programs that describe different (but similar) systems (4). A system based on reasoning from first principles can be easier to maintain because modifications to the design are fairly simple to implement. Structure and behavior specifications can be updated for each modified component rather than modifying the overall behavior of the entire system (2).

Another advantage of reasoning from first principles involves reasoning about novel faults. Since reasoning from first principles is not dependent on a catalog of observed malfunctioning behavior, it is possible to reason about a fault that has not occurred previously (2). Reasoning about novel faults is something experts are equipped to handle, but this type of experiential knowledge is difficult to encode in a diagnostic reasoning program simply because you can't inquire about malfunctioning behavior the expert has not dealt with.

There are several areas of experiential knowledge that can be encoded into a program that assist in developing a more complete diagnostic reasoning system. An expert is often aware of connectivity and adjacency not explicit in first principle knowledge. The expert also uses common sense reasoning and has the ability to reason outside a closed system domain. For example; if a system that contained exposed electrical circuits exhibited a fault after a rainstorm, the expert would tend to associate a malfunction with information about bad weather and check exposed circuits before performing a systematic search to isolate the malfunction. An expert can also develop a list of ordered categories of failure for each observed malfunction. By categories of failure we mean a list of possible search paths used to isolate a particular failure. In order to reason about a malfunction, the expert uses several pieces of information (e.g. similarities to other malfunction behavior, experience from similar systems, component failure history, external effects, knowledge about where and how to take measurements) and constructs a plan or measurement strategy to guide the troubleshooting process. The plan the expert has developed is essentially an ordered list of categories of failure. The expert also has the ability to know when an incorrect assumption has been made (i.e. an incorrect path has been followed) and how to regress and continue the troubleshooting process.(2)

Enumerating the advantages of reasoning from first principles and the advantages of reasoning from experiential knowledge has shown us that both are good approaches to automating diagnostic reasoning systems, however, in order to understand why both are not completely adequate as stand alone approaches we will now examine the disadvantages of each approach.

A program designed to reason from first principles will have difficulties constraining possible causes of failure. When reasoning solely from first principles, the troubleshooting process involves a systematic search of all possible paths that might lead to the malfunctioning component. There is seldom enough information to indicate a reasonable ordering of search paths, or to constrain the systematic search to a reasonable subset of search paths. In contrast to this, the expert rarely begins a troubleshooting process without constraining and ordering the possible search paths, this will allow the expert to reach a conclusion about a fault more rapidly than a system based entirely on first principle knowledge, if the fault can be reached by one of the paths the expert has selected. If the fault lies outside of the planned search paths the expert will be required to consult first principle knowledge located in some form of documentation, this will then increase the time necessary for the to expert to diagnose a fault.

Any time a program is required to reason with incomplete data, deficiencies occur. Many systems that are candidates for automation have incomplete, inaccurate, or unavailable documentation, this causes difficulties when attempting to reason using either first principles or experiential knowledge. An automated system reasoning from first principle knowledge is only as accurate as the documentation that was used to build it, however, experiential knowledge that can be encoded in programs is quite often based on reasoning about a system where system documentation is incomplete.

A program designed to reason from experiential knowledge is based on empirical associations and is usually difficult to construct. The process of attempting to model a person is long and there is often no way to know if a complete set of knowledge has been extracted from the expert. When developing a program that reasons from experiential knowledge, the developer often must choose between experts who have different opinions about how to solve problems

and different ideas about the cause of faults that have occurred previously. When making these choices the developer will possibly limit the efficiency of the program. In contrast, a program that reasons from first principles is not limited to a set of predefined solutions and can reason about a fault without any limitations on the conclusions that can be reached.

Programs that reason from experiential knowledge are restricted by an expert's sample cases, and restricted to the domain there were intended for. As mentioned earlier, the process the expert uses to reason about a novel fault is not an area of reasoning that can be automated using todays technology. Therefore, reasoning strictly from experiential knowledge restricts the program to reasoning only about cases elaborated on by the expert.(2)

5. COMBINING APPROACHES

From the discussion so far, it should be obvious to the reader (as it might have been prior to this reading, since we are not pretending to present unique or unusual ideas but simply our understanding of the requirements for a diagnostic system in the field) it is our view that both first principle and experiential reasoning are necessary for a robust diagnostic program operating outside the laboratory environment. We offer the following examples from the Space Program to illustrate the nature of this mixture of reasoning approaches.

For the handling of malfunctions on the Shuttle a set of procedures called Malfunction Procedures (MALF's) are generated for the crew to follow after they observe some abnormal behavior in a system. These MALF's, which take the form of a decision tree, embody both the first principle and experiential reasoning discussed above. They are developed to take into consideration a wide variety of possible system failures in a variety of system contexts in order to plan well understood and safe malfunction responses. However, a complete specification of all system failures is not practicable (let alone possible) and where appropriate the crew is directed to simply contact Mission Control and report their observations. The personnel on the ground in turn have a more detailed set of MALF's that are used as above, but also have a complete set of documents that represent a first principle understanding of the system and are frequently referred to during diagnosis. Additionally, these personnel have a wealth of experience with the system in question, and the world in general, which they draw upon for reasoning that must go beyond an understanding available from the documentation only.

As an example of extra-system knowledge being brought to bear on system diagnosis we can look at the actual experience of a Manned Maneuvering Unit pilot on the way back to the Shuttle during STS 41C. He observed that the relative size of the Shuttle was growing faster than appropriate for an approach at constant velocity. This indicated that there was a definite relative acceleration along the line-of-sight to the Shuttle. The situation occurred within an agreed context between the individuals involved in the approach, the MMU pilot would be the only individual imparting a relative velocity between the vehicles. One possible explanation for this, that can be derived from an in-depth understanding of the MMU's structure and behavior, is a stuck-on MMU thruster; a fairly serious malfunction. According to good training and his understanding of the situation context, the pilot then began to correct for the unintended acceleration by slowing the MMU and proceeded to inform Mission Control of his situation. At that time he learned that the Shuttle pilot was actually accelerating the Shuttle toward him. With this new contextual information the pilot then could explain his observables in terms of these extra-system factors.

Given this view of the nature of a major portion of the reasoning needed for a comprehensive diagnostic system, how does one go about developing such a program. It as our view that the best approach would be to divide the needed reasoning tasks into those tasks and subtasks directly accomplished via the first principle approach and those tasks subtasks that are better handled by modeling the behavior of the expert directly.

6. KNOWLEDGE ACQUISITION

Dividing reasoning tasks into sections corresponding to the two approaches has some effects on the knowledge acquisition process that are advantageous to the program construction process. Combining the approaches permits the systematic enumeration of a large portion of the knowledge needed to construct a diagnostic reasoning program; this facilitates defining the program's scope, completeness and competence, and assists in bounding, controlling and guiding the knowledge acquisition process. In this paper we consider knowledge acquisition to be acquisition of all knowledge that will be incorporated into a program; this includes knowledge from document sources as well as knowledge from an expert(s).

First principle knowledge to be acquired from documentation, schematics, drawings, etc. can be stated explicitly by enumerating components, correct component behavior, component connectivity and effects, and description of observables. Although extracting knowledge from documented material is not terribly difficult, (especially when compared to extracting knowledge from a human) it is still a complex task and can be simplified by extracting the needed knowledge using a systematic method (2). Once the process of gathering first principle knowledge is completed, the task of extracting knowledge from the expert is reduced because you are not relying on the expert to provide

you with necessary design knowledge about structure and behavior.

Unfortunately, the approach to enumerating knowledge required from the expert is not as systematic as with first principle knowledge gathering; however, a set of goals can be generated to help elicit knowledge from the expert. The list of goals are intended to guide the interaction with the expert toward the elicitation of strategies used to perform the monitoring task, detection task, isolation task, and recovery task (discussed in section 3). Knowledge acquired from this effort includes (but is not limited to) determining components not deseribed in documentation, implied connections not explicit in documentation, environmental effects on components and component behavior, previous failures, failure trends, untestable observables, information about components inferred from measurements of other components (8), functional leveling (the amount of structural and behavior detail needed to model components varies with level of abstraction), and ordering of categories of failure (discussed in section 4).

When attempting to acquire knowledge from experts, it should be realized that they tend to have a variety a models that are used during a diagnosis. Gaining insight into what model the expert is using and how the expert developed the model can reveal valuable information about how the expert performs a troubleshooting task and information about different levels of structural and behavioral detail needed to reason about a fault (4). For example; when diagnosing a car that won't start, you would rarely begin by reasoning about a wiring harness diagram and its connections to the ignition system. Rather, you would most likely think of the wiring harness as a 'black box' until there is an indication that the fault lies within the wiring harness. Unfortunately, an expert's selection and development of models is a process that is not well understood (1) making the extraction of these models a difficult and involved part of knowledge acquisition. Since experts are rarely explicit about the models they use, it is advisable to construct scenarios the expert can reason about and develop the model from strategy paths the expert uses (8).

Since we are combining two approaches, the question arises concerning how much knowledge from each area should be included in the program and when is the knowledge acquisition job complete? As with any project, the desire is to make the program as complete as possible, however, the underlying issue is still the balancing of completeness (detail of structure and behavior) and the need to constrain the search paths (categories of failure) the program will reason about when diagnosing a fault. A program that cannot respond quickly to a malfunction will certainly be unacceptable in certain domains, as such each project must determine the overall requirements of the program before the knowledge acquisition process is initiated (8).

7. CONCLUSIONS

Completeness, efficiency and autonomy are requirements for future diagnostic reasoning systems. Methods for automating diagnostic reasoning systems include diagnosis from first principles (i.e. reasoning from a thorough description of structure and behavior) and diagnosis from experiential knowledge (i.e. reasoning from a set of examples obtained from experts), however, implementation of either as a single reasoning method fails to meet these requirements. The approach of combining reasoning from first principles and reasoning from experiential knowledge does address the requirements discussed above and can possibly ease some of the difficulties associated with knowledge acquisition by allowing developers to systematically enumerate a portion of the knowledge necessary to build the diagnosis program. The ability to enumerate knowledge systematically facilitates defining the program's scope, completeness, and competence and assists in bounding, controlling, and guiding the knowledge acquisition process.

REFERENCES

1. Davis, R., "Robustness and Transparency in Intelligent Systems", Symposium on Human Factors Needs in Space Station Design, Washington, DC, Jan 1987.

2. Davis, R., "Diagnostic Reasoning Based on Structure and Behavior", Artificial Intelligence, Elsevier Science Publishers B.V. (North-Holland), 24, 1984, 347-410.

3. Davis, R., Buchanan, B. and Shortliffe, E., "Production Rules as a Representation in a Knowledge-Based Consultation System", Artificial Intelligence, Elsevier Science Publishers B.V. (North-Holland), 8, 1977, 15-45.

4. De Jong, K., "Knowledge Acquisition for Fault Isolation Expert Systems", Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, Nov 1986.

5. Hamscher, W., and Davis, R., "Diagnosing Circuits with State: An Inherently Underconstrained Problem", Proceedings Of National Conference on AI, Austin, TX, August 1984.

6. Reiter, R., "A Theory of Diagnosis from First Principles", Artificial Intelligence, Elsevier Science Publishers B.V. (North-Holland), 32, 1987, 57-95.

7. Shortliffe, E., Computer-Based Medical Consultations: Mycin, American Elsevier, New York, 1976.

8. Woods, D. and Hollnagel, E., "Mapping Cognitive Demands and Activities in Complex Problem Solving Worlds", Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, Nov 1986.